ICSC
Centro Nazionale di Ricerca in HPC,
Big Data and Quantum Computing

Università degli Studi di Padova

FPGA-related activities at UNIPD: TENET and FEROCE

Andrea Triossi on behalf of Boost Lab

University of Padova

"This work is partially supported by ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU".

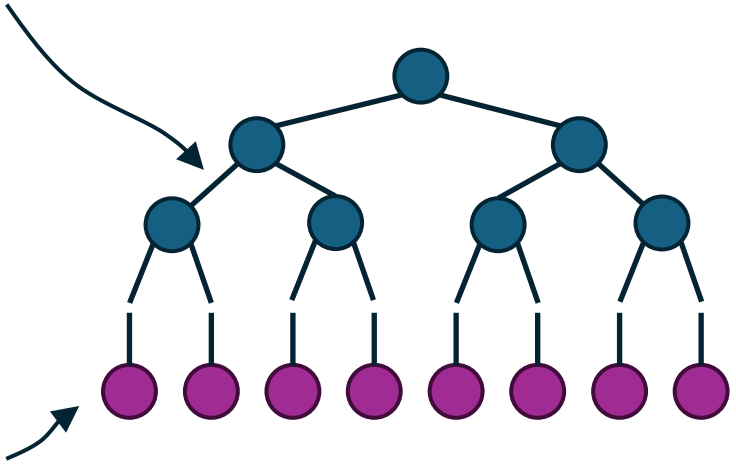
Tensor Networks

- Collection of tensors connected by contractions
- Intuitive graphical language
  - Tensors are notated by shapes
  - Indices are notated by lines emanating from these shapes
  - Connecting two index lines implies a contraction

- Approximate arbitrarily complex many-body quantum systems preserving its most important properties

Tree Tensor Networks for ML

- TN with a tree structure
- **Bond dimension** (χ_l) controls the expressivity of the TTN



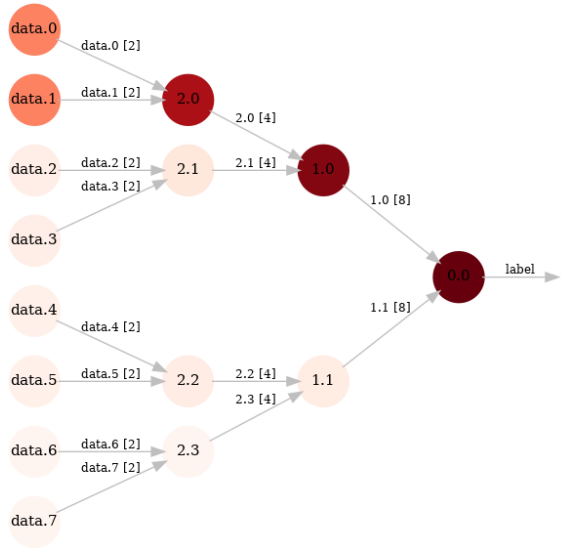
- **Input features** (N) are mapped in a higher dimensional space (D)
- Number of layers $L = \log_2 N$
- χ_l ideally scales with the layer as D^{2^l}
- Reduce the complexity of the problem artificially forcing $\chi_l = \min(D^{2^l}, \chi_0)$

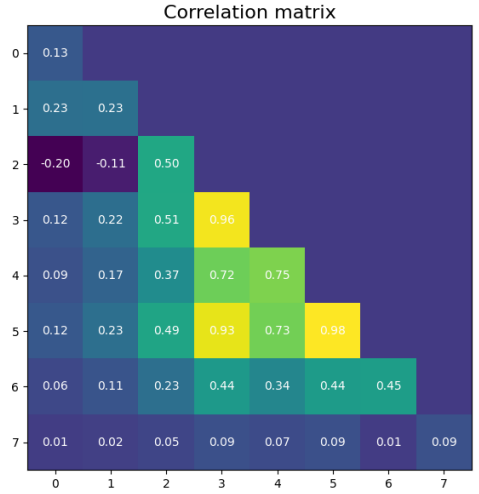
Methods

- Goal: binary classifier with ultra-low latency
- Training is done in software
- Weights are loaded in the dedicated hardware (FPGA) for inference
- Inference consists of only linear transformations
- FPGA is a programmable devices that combines an array of combinatorial logic blocks with a mesh of interconnections
  - Look-up tables, storage elements, fast carry chains
  - Dedicated hardware for specific functions (RAM, PLL, Ser/Des)
  - Digital Signal Processor (DSP) for arithmetic functions (adder, multiplier, accumulator)
  - Hardware Description Language (HDL) for circuit description
  - High degree of parallelization but limited resources

Training

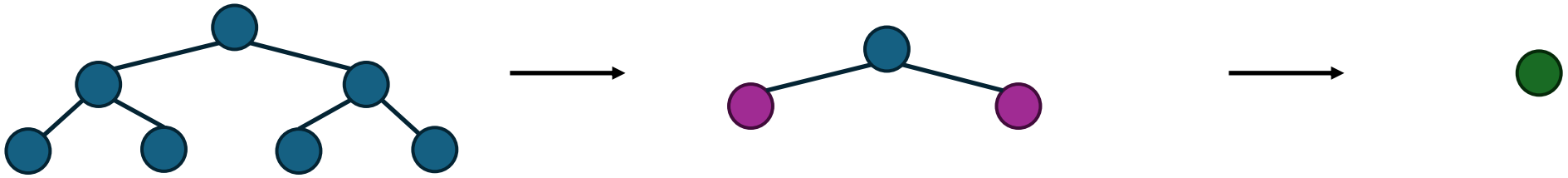
- Pythonic custom classes for representing TTN as a Torch NN module
- Tested on three datasets
  - Iris – N=4, D=2, χ_0=4 ⟶ 99% Accuracy
  - Titanic – N=8, D=2, χ_0=3,4,8,16⟶ 79% Accuracy
  - LHCb $b/\bar{b}$ tagging – N=16, D=2, χ_0=8,16⟶ 62 % Accuracy
- Methods to measure physical quantities
  - Entropy of each link
  - Correlation between features
- Ranking of the input features
  - Complexity reduction of the TTN





Tensor contraction

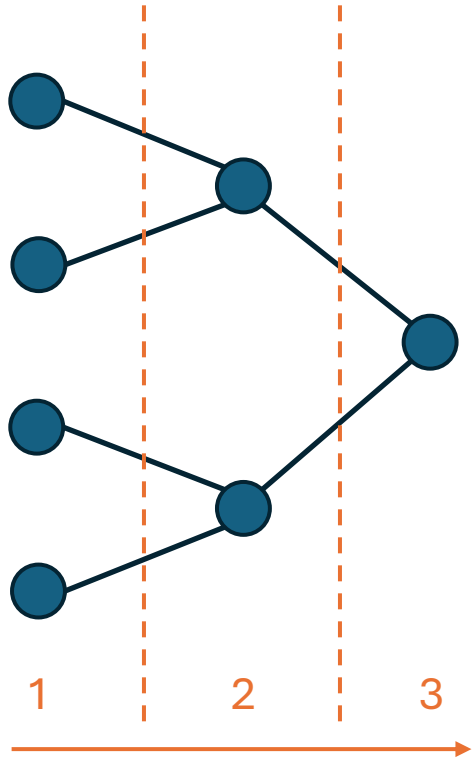
- Inference means to contract the full tree



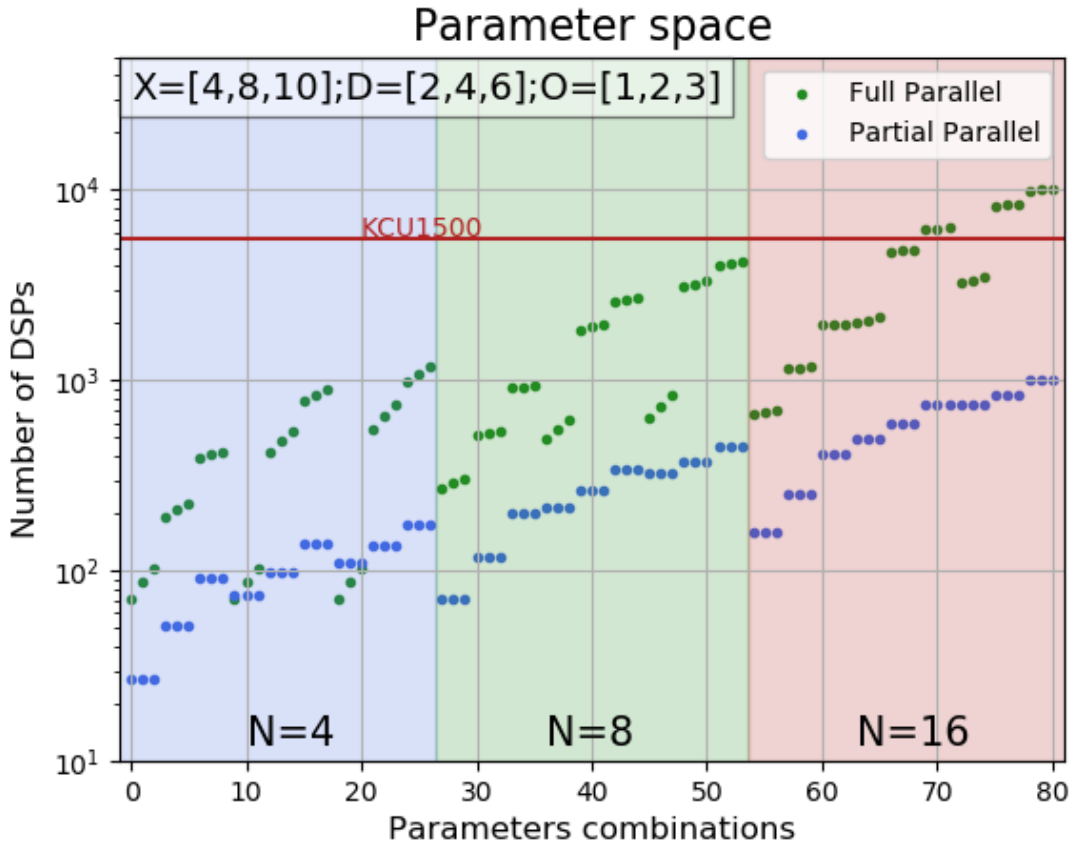
- Contraction operation in an order-3 tensor: $c_k = \sum_{ij} a_i b_j W_{ijk}$
- DSP has two inputs ⟶ two multiplication stages are needed
- We explored two degree of parallelism for the contraction
  - Full parallel – where the exploited number of DSPs is maximal
  - Partially parallel – where we introduce a reuse of the DSPs

Execution flow

- Feature map implemented in LUTs
- Node computation is independent
- Each layer is computed in parallel
- Layers are fully pipelined
- After full contraction, output is a scalar



Resources

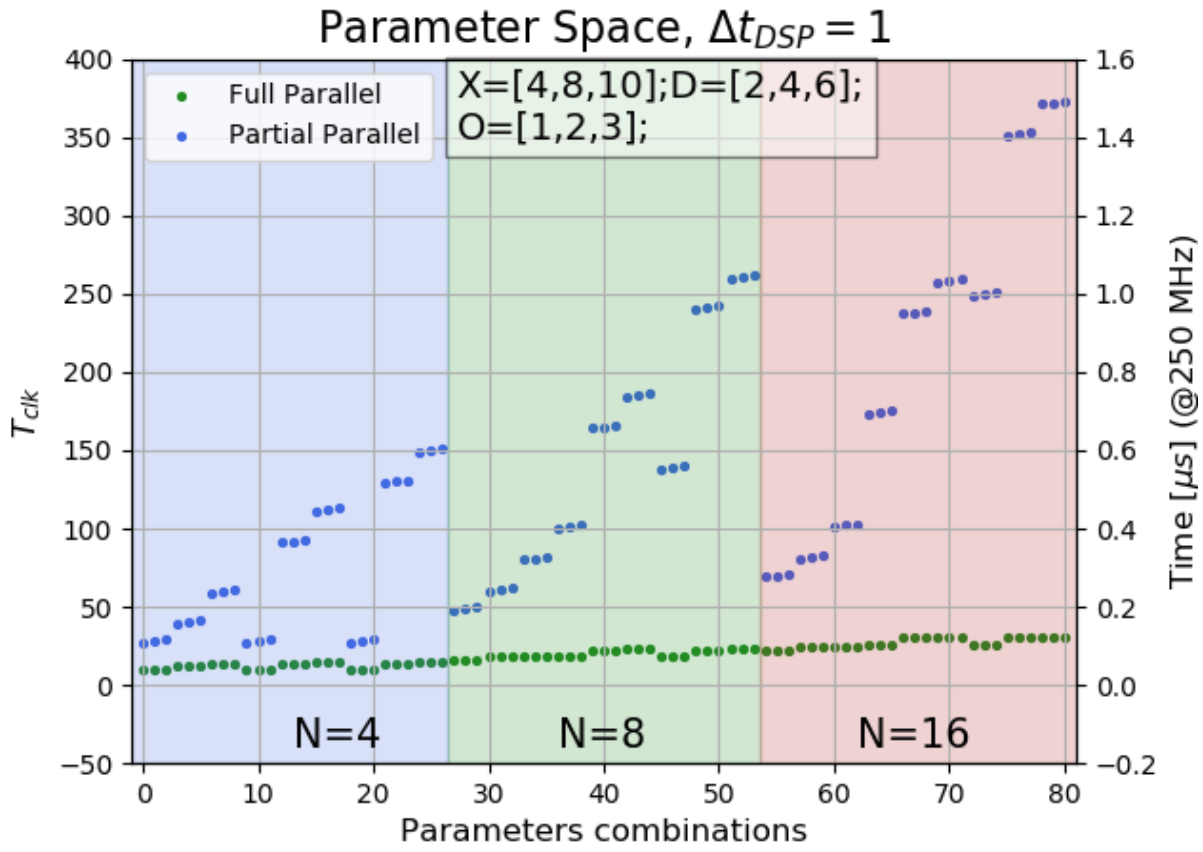


Total number of DSPs

$$\sum_{l=1}^{L} \chi_{l-1}^2 (\chi_l + 1) \frac{N}{2^l}$$

$$\sum_{l=1}^{L} (\chi_{l-1}^2 + 1) \frac{N}{2^l}$$

Latency



Total latency

$$\Delta_{DSP} \sum_{l=1}^{L} 2 + \log_2(\chi_{l-1}^2)$$

$$\Delta_{DSP} \sum_{l=1}^{N} \chi_{l-1}^2 + \chi_l + 1$$

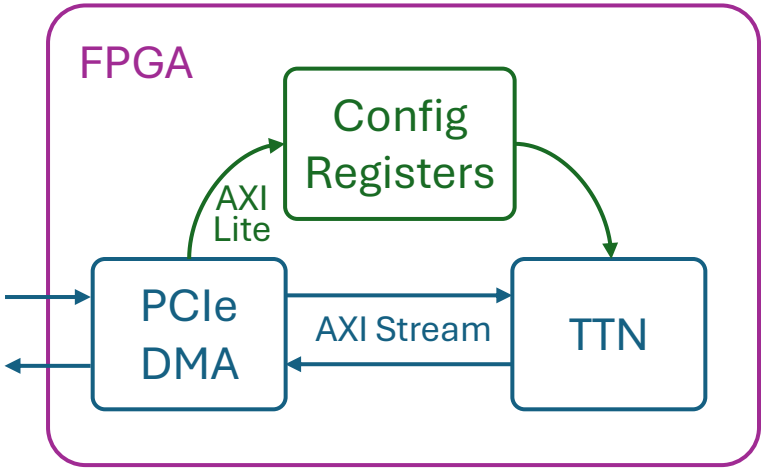
Hardware setup

Server

PCIe

KCU1500
Kintex Ultrascale

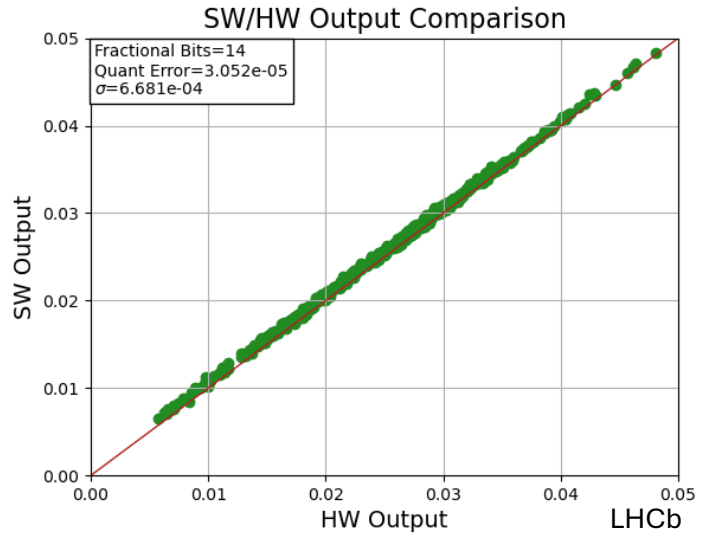
- TTN deployed on hardware accelerator
- Offloading of the TTN inference
- Application on top of Xilinx DMA drivers

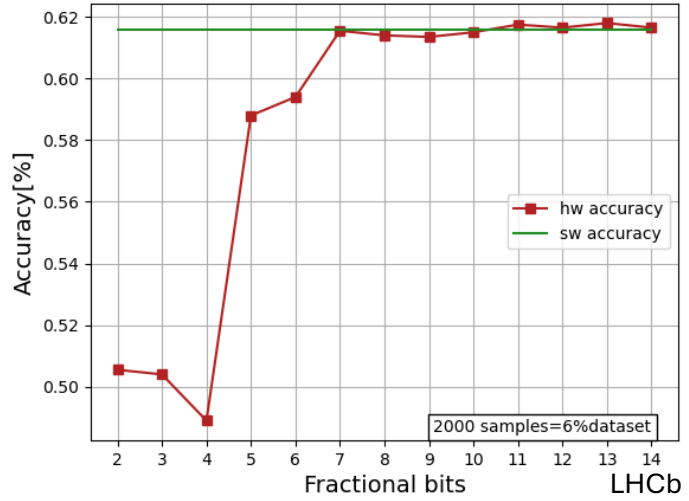


Results

- SW/HW comparison
- Classification accuracy vs number of bits
- Actual resources and latency agree with expectations

Dataset	TTN	DSP	BRAM	Latency
Iris	[2,4,1] PP	1%	2%	108 ns
Titanic	[2,4,8,1] FP	8%	19%	72 ns
LHCb	[2,4,8,8,1] FP	36.5%	84%	104 ns



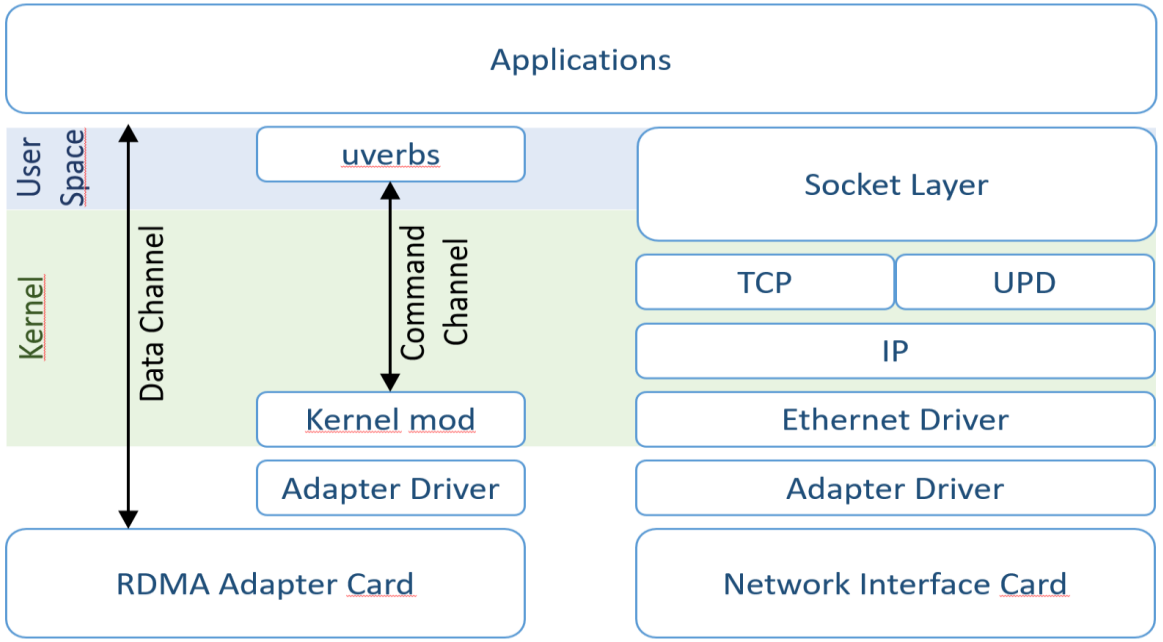


Status and perspective works

- Project presented at ICHEP24
- Paper submitted to «IEEE Transactions on Emerging Topics in Computing»
  - Pre-print available in: arXiv
- Next steps:
  - Integration in trigger systems
  - Move to higher language description (HLS/C++)
    - Exploiting Versal AI Engines
  - Explore possible use cases beyond HEP

Objectives

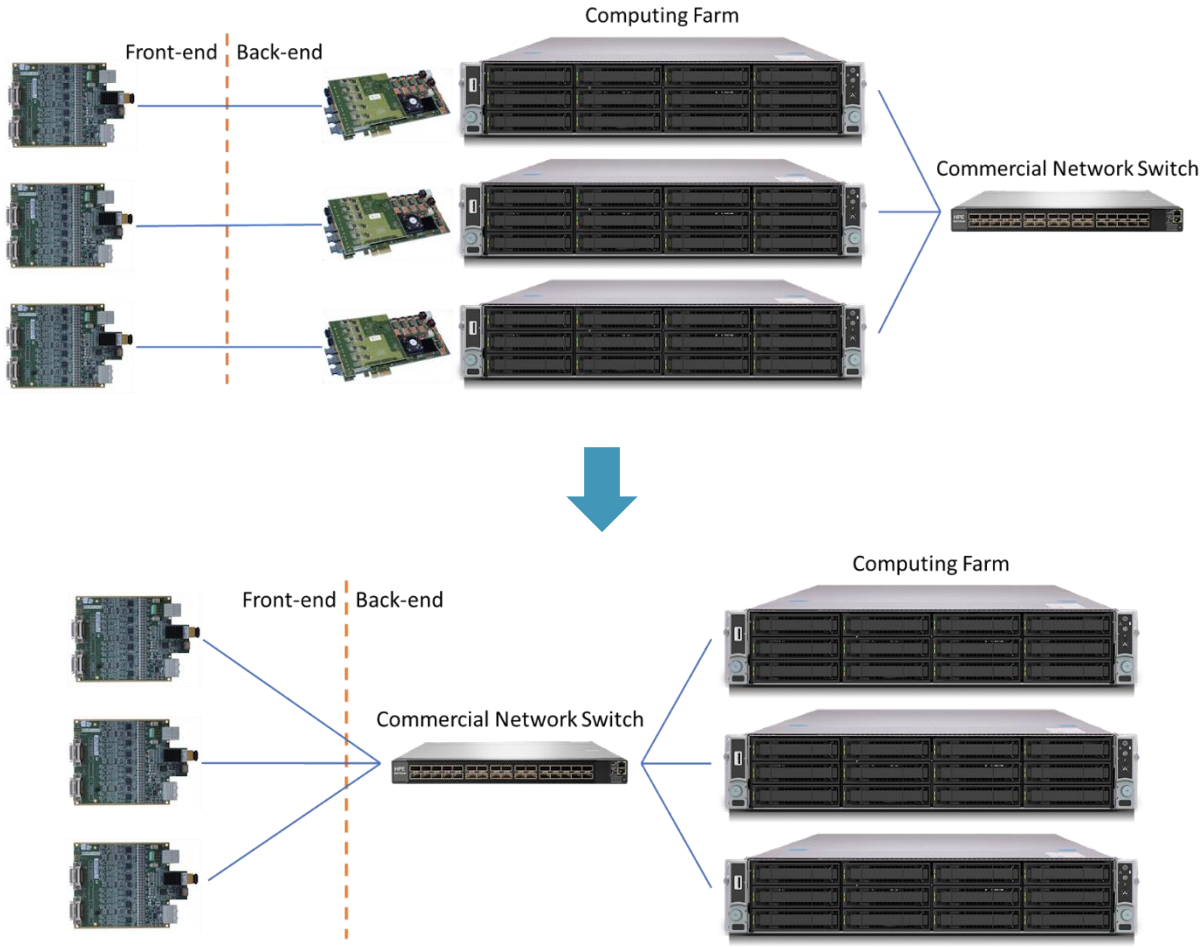
- Processing power is important as an efficient data movement
- In a DAQ system a large fraction of CPU is engaged in networking
  - Data manipulation (several copies)
  - Latency increase and throughput reduction
- Zero-copy is obtained by adding RDMA layer to the network stack



- FEROCE wants to move the adoption of the network protocol to the data producer
  - Front-end initiates the RDMA transfer
  - No point-to-point connection between front-end and back-end
  - Dynamical switching routing according to node availability

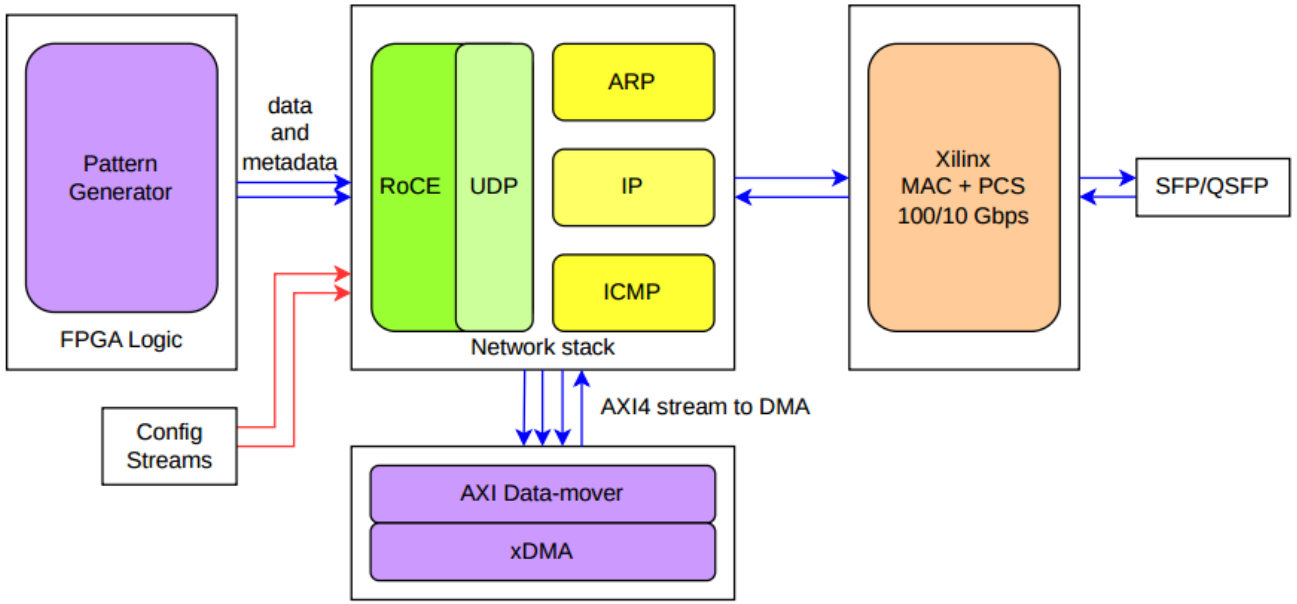
Methodology

- Several network stacks implementing RDMA
  - InfiniBand, RoCE, iWARP…
- RoCE (RDMA over Converged Ethernet)
  - Based on Ethernet networks
  - Industry-standard
  - Multi-vendor ecosystem
  - RoCE v2 packet switching (layer 2 and 3)
- FPGA are already used for implementing network stacks
  - Data center
  - ATLAS



Study of the exiting libraries

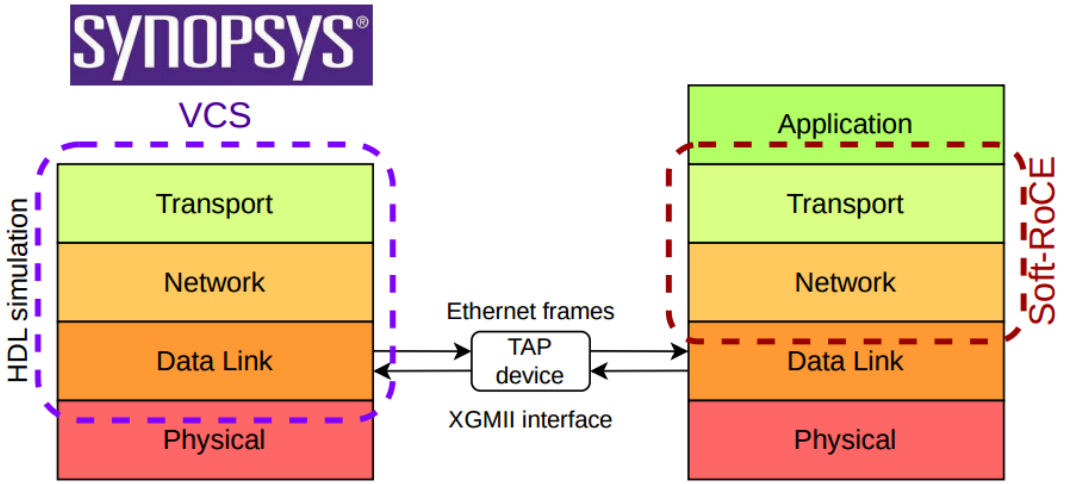
- Open-source libraries
  - ETH Zurich Network Stack
    - Entirely written in HLS
    - 10/100 Gbps via Xilinx 10G and 100G MAC IPs
    - xDMA, DDR4 memory and recently HBM support



- This module writes/reads data to/from the Host machine's memory through the xDMA
- An AXI data-mover is used to translate the AXI4 stream into AXI memory mapped (and viceversa)
- Queue Pair's information is exchanged using AXI stream ports
- A debug port is present to send data directly from the FPGA logic TX DATA port

Testing of UDP, TCP and RoCE V2

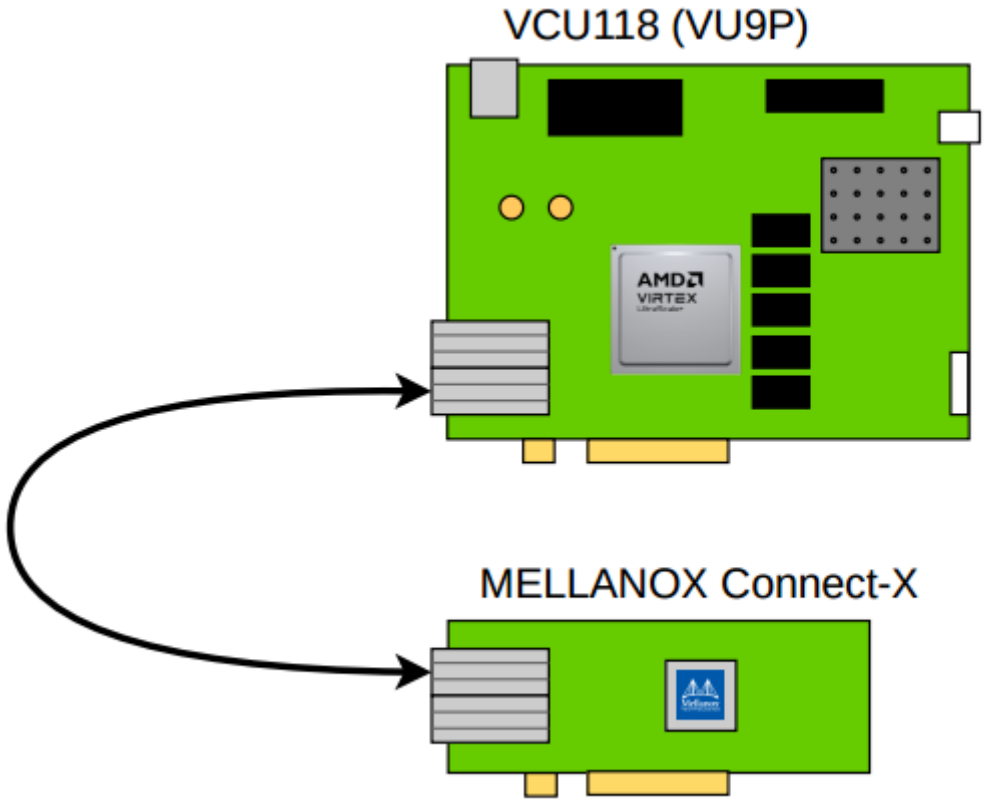
- Dynamic VCS simulation
  - TUN/TAP
  - DPI-C (Direct Programming Interface) used to link C functions with RTL simulation
  - Modifications in MAC and in DDR interface
  - RDMA WRITE tested successfully with a 10G MAC
  - RoCE firmware simulator sends data to Soft-RoCE end-point



- Results presented at two international conferences TIPP and TWEPP
  - Proceeding here

Hardware implementation

- UDP and TCP stacks deployment on VCU118
  - Good bandwidth results for TCP (> 60Gbps with two connections)
  - Hard to close the timing (even in HLS doesn't close)
  - Porting to Vivado 22.2 was needed (intelligent runs)
- RoCE v2 stacks deployment on VCU118
  - Several issues were found before successfully transmit data (also present in simulation)
  - Occupancy:

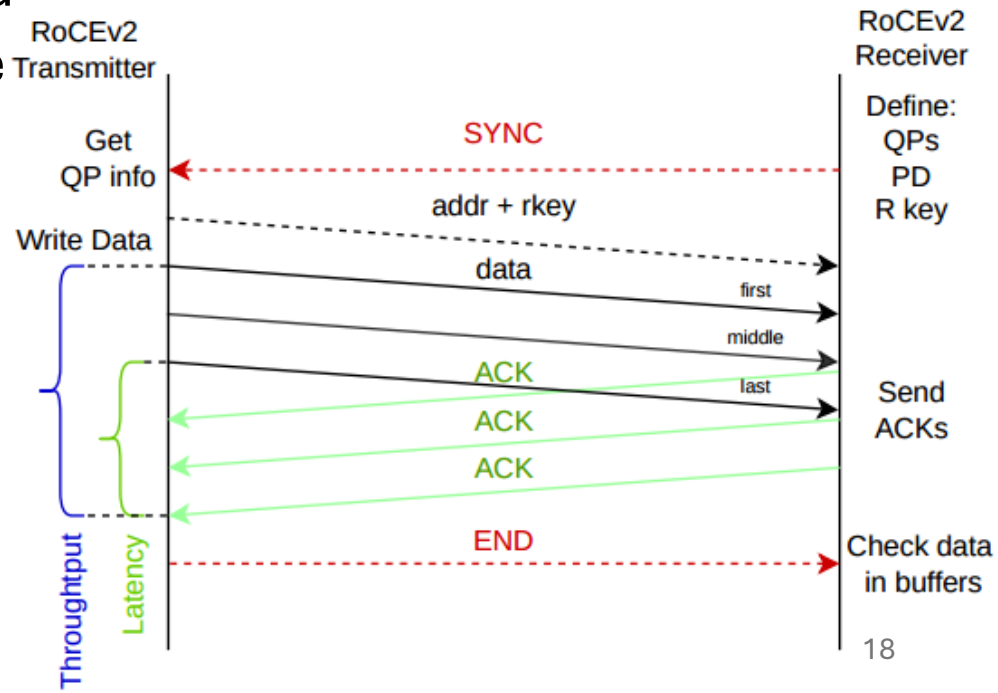


RoCEv2 (no retrans)			TX only estimate (no retrans)		
LUT [k]	FF [k]	BRAM	LUT [k]	FF [k]	BRAM
38	49	93	~ 15	~ 15	~ 30

Latency and throughput

- Latency is computed averaging the time from each packet's transfer start and its acknowledge (round trip)
- Throughput is computed considering the payload size and the time from first packet sent and last acknowledge received
- First transfer in the QP must be small (warm-up), it's needed to for caching the QP info (NIC side), otherwise packet loss is observed
- After that, no packet loss is observed, need more tests though

Transfer size	Throughput [Gbps]	Latency [μs]
8 kB	30.7	1.6
32 kB	62.8	2.3
64 kB	73.3	3.1
128 kB	80.5	3.3
512 kB	92.9	12.4
5.12 MB	95.8	12.4
51.2 MB	96.0	12.4
512 MB	96.0	12.4



Light-ROCE Tx Module

- Written at RTL (Verilog)
- Only RDMA WRITE with immediate for completion of the transfer
- Open-source network stack for low level layers (UDP, IP, ARP…)
- Slow control based on UDP (setting QP, etc.)
- Tested in dynamic simulation towards soft-ROCE
- Main GitHub code
  - Supported speeds 10G/25G/100G
  - FAST CRC32
  - Support scripts for packet verification
- Talk accepted at CHEP 2024
- Next steps
  - Add re-transmission module (depending on 100G congestion test)
  - Rewrite RoCE Rx module at RTL only to decode ACK/NACK/CNP packets
  - Porting of the ROCE network stack on Microchip FPGA (evaluation board received)

Tests on Light-ROCE

- Occupancy of 10G/25G fw
- Congestion tests on a 10G/25G network
  - Two senders / one receiver link saturation
  - Flow Control (FC) or RoceV2 congestion management (DCQCN) throttle the sender without triggering retransmission
- Throughput and latency measurements for the VCU118 at 10G and 25G
- Next steps
  - Repeat tests on a 100G network

	LUTs (K)	FFs (K)	BRAMs
RoCE + ICRC	5.4	6.3	5
Light-ROCE	14	16	11
Full ETH stack	38	49	93

	Latency (us)	Throughput (Gbps)
Point to point @ 10G	4.6	9.3
Congestion @ 10G (PMTU 2048)	7.8	9.3
Congestion @ 10G (PMTU 4096)	13.2	9.64
Point to point @ 25G	4.5	24.1
Congestion @ 25G	20	24.1

Prospective applications

- CMS
  - L1T Scouting is a project aiming at acquiring the L1 primitives at the full bunch crossing rate
  - It is meant for HL-LHC but a demonstrator based on commercial electronic is already deployed
  - At present as DAQ link it adopts a light version of the TCP/IP protocol at 100G ⟶ move to light ROCE